

**REMARKS**

Claims 9-14, 16-26 and 30 are pending in the patent application.

The Examiner has objected to Claims 9-26 based on informalities in independent Claims 9 and 12. Applicants have amended the preambles of Claims 9 and 12 to address the objections.

The Examiner has objected to the Specification for failing to provide antecedent basis for "dynamically" and "directly". Applicants have amended the claim language and obviated any objection to the Specification as further discussed below.

The Examiner has rejected Claims 9-26 and 30 under 35 USC 112 for the terms "dynamically", "separate" and "directly". Applicants first note that Claim 15 was canceled by the last Amendment.

With regard to the term "separate", Applicants have amended the claim language, as further detailed below, to make it clear that a first trusted connection is established between the server and the terminal (which finds support in the Specification, for example at page 9, lines 18-21, page 12, lines 4-5, page 16, lines 17-20 and page 19, lines 7-11)

**SZ998-041****-9-**

and that a second trusted connection is established between the server and the user device (which finds support in the Specification, for example at page 10, line 3, page 12, lines 8-9, page 17, line 3 and page 19, lines 16-17). Applicants respectfully assert that, although the Specification did not expressly use the term "separate" in detailing the connections, it is abundantly clear to one having skill in the art that the two trusted connections are separate. The Examiner has pointed to the language in the Specification that states that the second trusted connection may be tunneled through the first trusted connection. Applicants supply herewith several references which define the term "tunneling" and describe tunnel connections. It is clear from a review of those references that communications along a tunneled connection between two entities, (e.g., between the server and the user device), are not communicated **TO** any other entity (i.e., the terminal) which may share the connecting route. Tunneled communications are secure and directed only to the intended recipient, regardless of the existence of intermediate entities along the route. It is clear from Applicants teachings and claim language that a message sent along a second trusted connection between a server and a user device is delivered

SZ998-041

-10-

without delivery of same to the terminal. In that regard, the trusted connections are "separate". While Applicants have removed the term "separate" from the claims, Applicants still maintain that the delivery of a message along one trusted connection from the server to the user device does not result in delivery of the message along the other trusted connection from the server to the terminal. Applicants believe that the claim language as amended overcomes the Examiner's objections and rejections related to the use of "separate" and "directly".

With regard to the user of the term "dynamically", Applicants have amended the claims to remove the term and have inserted the phrase "in response to user input at the terminal" which finds support in the Specification at page 9, line 15, page 16, line 15 and page 19, lines 3-4. The terminal authentication message is generated by the server in response to user input to the terminal. Applicants believe that the amendments overcome the Examiner's objections and rejections related to the use of the term "dynamically".

The Examiner has rejected Claims 18 and 22 since the claim from which they depended has been canceled. Amendments have been made to correct the dependencies.

SZ998-041

-11-

is authorized to access the server. In all claimed embodiments of the invention, two trusted connections are established, a first between the server and the terminal and a second between the server and the user device. In response to user input at the terminal, that input being insertion of a user device, input to the terminal, etc., the server first authenticates the terminal. Once the terminal has been authenticated, the server communicates that information along the second connection between the user device and the server, without communicating that information along the connection between the server and the terminal. As discussed above, even if the two trusted connections share the same physical path, delivery along one connection does not comprise delivery along the other connection. The server either communicates that information directly to the user by display at the user device, or communicates that information to the user by notifying the user device whereupon the user device causes the terminal to display the information to the user, when the user has a device that does not have display capabilities. Applicants respectfully assert that none of the cited prior art teaches or suggests a server communicating terminal authentication information directly to the user device along a trusted

SZ998-041

-13-

The Examiner has rejected Claims 9-14, 16-18, 21-22, 25-26 and 30 under 35 USC 103 as unpatentable over the combined teachings of Merritt, Manduley, Hoss and Limsico; Claim 19 under 35 USC 103 as unpatentable over the combined teachings of Merritt, Manduley, Hoss and Limsico and further in view of Abraham; Claim 20 under 35 USC 103 as unpatentable over the combined teachings of Merritt, Manduley, Hoss and Limsico further in view of Lessin; and Claims 23 and 24 under 35 USC 103 as unpatentable over the combined teachings of Merritt, Manduley, Hoss and Limsico and further in view of Schneier. For the reasons set forth below, Applicants respectfully assert that the claims, as amended, are patentable over the cited art.

The present invention teaches and claims a device, terminal, server, program storage device, and method for establishing trustworthy connections among a user, with or without a device inserted at a terminal, a terminal, and a server. Specifically, the user must know that the terminal is trusted by the server before the user will release any sensitive information to the terminal. Similarly, the server must know that the terminal seeking access to it is authentic. The server may also engage in an exchange to determine if the user, of a user device or of the terminal,

SZ998-041

-12-

connection between the server and the user device without communicating that information to the terminal along the trusted connection between the server and the terminal. Applicants also assert that none of the prior art teaches or suggests that terminal authentication information be communicated to the user, whereupon the user or user device provides information to the terminal for the terminal to dynamically create a user-specific authenticity output message for display to the user.

The primary reference cited against the present application is the Merritt patent. The Merritt patent teaches a method for authenticating a terminal whereby a terminal contacts the server, followed by the server and the terminal engaging in an authentication process, referred to as a "two-way challenge-response" (Col. 4, lines 57-64). Once the server and the terminal have mutually authenticated themselves to each other, the terminal sends the user's account information to the server, the server retrieves a user-specific personal security phrase ("PSP") from its storage, and the server sends the PSP to the terminal. The terminal displays the PSP to the user, prompting the user to verify that the PSP is correct, preferably by user entry of a personal identification number (PIN) (see: Col. 6, lines

SZ998-041

-14-

21-36). Under the Merritt method, the server does not generate an authenticity output message and does not communicate a generated authenticity output message to the user along a connection which is separate from the connection between the host and the terminal. Rather, under the Merritt system, the server authenticates itself and sends that information to the terminal while the terminal authenticates itself and sends that information to the server. The Merritt server does not authenticate the terminal, but authenticates itself (i.e., the server) to the terminal (Col. 4, lines 60-64). The server does not authenticate the terminal and does not generate any authenticity output message regarding the authenticity of the terminal. Further, under Merritt, the user does not receive any authentication information, from either the terminal or the server. The user simply gets prompted with his PSP. The user does not have a separate connection with the host and does not receive terminal authentication information from the host.

With specific reference to the language of independent Claim 9, Applicants respectfully assert that the Merritt patent does not teach or suggest the invention as claimed. The Merritt system does not teach or suggest that the server

SZ998-041

-15-

has a communication component for establishing and conducting communications with a terminal along a first trusted connection and for establishing and conducting communications with a user along a second trusted connection. Merritt provides one communication line, 9 of Fig. 1, between the host/server and the terminal. Merritt does not teach or suggest that the user have a trusted connection with the server. Applicants contend that communicating along the one connection, 9 of Fig. 1, between the host/server and the terminal does not teach or suggest a second trusted connection between the server and the user, which second trusted connection is different from the first trusted connection between the server and the terminal. In response to the 112 rejection of this language, Applicants aver that throughout the Specification, different first and second authenticated trusted connections are clearly taught (see: S-T and S-D on page 9, step 3 through page 10, step 5; c1 and c2 at page 12, lines 3-9; c1 and c2 on page 13, lines 14-18; and S-T and S-D on pages 16-17, etc.). Applicants contend that the Specification is clearly teaching separate connections, as would be clear to one having skill in the relevant art from a reading of the teachings referenced above.

SZ998-041

-16-



The claim language of Claim 9 also expressly recites that the server has at least one authentication component for verifying the authenticity of the terminal. According to the teachings found in Merritt at Col. 4, lines 60-64, however, the server authenticates *itself* to the ATM terminal (in step 340 of Fig. 3) but does not authenticate the terminal, per se. While Fig. 1 of Merritt does illustrate a comparator component and RN generator, Merritt does not teach that the components comprise an authentication component for verifying the authenticity of a terminal. The Examiner additionally cites Fig. 3, element 315 against the authenticity component. What is illustrated at 315 of Fig. 3 is the process step of the two-way challenge-response process. Element 315 does not illustrate a server authentication component. Finally, the Examiner cites the passage found from Col. 2, lines 10-14 against the claimed at least one authentication component. The cited passage states that there is a need to authenticate a terminal to a user. Neither the cited passage nor the ensuing Merritt teachings, however, expressly teach that the terminal is authenticated by the server.

The independent Claim 9 further recites a message generation component for generating at least one terminal

authenticity output message for delivery to the user along the second trusted connection. Applicants contend that Merritt does not teach or suggest that its host server has a message generation component that generates a terminal authenticity output message in response to user input at a terminal. The Examiner cites element 3 of Fig. 1 as showing both a message generation component and a storage location (see: the top of page 9 of the Office Action). What Fig. 1, element 3 illustrates is a database. The only teachings of the host accessing that database are found at Col. 6, lines 22-23 and at Col. 7, lines 5-7 where the host retrieves the PSP and account information from the database. Applicants argue that it is clear that the Merritt element 3 database is not a message generation component but is simply a storage location. The Examiner also appears to analogize retrieving and displaying the PSP to the generation and display of a terminal authenticity output message indicating that the terminal has been authenticated. Applicants assert that Merritt does not teach or suggest that the PSP is a terminal authenticity message. The PSP is a retrieved user identifier. Applicants further reiterate that the PSP is delivered from the host to the terminal along the single Merritt connection. The PSP is not a terminal authenticity

SZ998-041

-18-

message which is send to the user along a second trusted connection between the server and the user device without delivery to the terminal. There is no teaching or suggestion in Merritt of a second trusted connection between the host and the user along which an authenticity message could be communicated.

Applicants maintain that the Merritt patent does not teach or suggest the steps of establishing a first authenticated trusted connection between the server and the terminal upon authenticating the terminal and of establishing a second trusted connection between the server and the user device. Merritt does not teach or suggest multiple connections. Applicants further assert that Merritt does not teach or suggest any communications between the host and the user that do not involve the ATM terminal. The Examiner, on page 3 of the Office Action, cites reference numeral 380 against the second trusted connection. However, reference numeral 380 illustrates the step of "ATM Communicates PSP to Customer". Clearly the ATM displaying the PSP to the user is not the same as or suggestive of the server communicating a terminal authenticity output message to the user along a second trusted connection between the server and the user device without delivery to the terminal

SZ998-041

-19-

along a first trusted connection between the server and the terminal. Rather, the Merritt ATM is displaying information which was received by the terminal from the host along the only connection.

Applicants further assert that the additionally cited Manduley patent does not provide the teachings which are missing from the Merritt patent. The Examiner acknowledges that the Merritt patent does not teach or suggest providing a terminal authenticity message to the device. The Manduley patent teaches a method for assuring that the user is actually in possession of the card. The invention as set forth in independent claim 12 expressly recites the server providing a terminal authenticity message to the device via the established second trusted connection. As claimed, the user device is being provided with confirmation that the terminal has been authenticated. User authentication is not being claimed. Moreover, sending terminal authentication information directly from a server to a user device along a connection which is separate from the connection between the terminal and the server, thereby eliminating the possibility of a terminal interfering with or falsely generating a terminal authentication message, is not taught or suggested by the Manduley device display. Neither Manduley nor

**SZ998-041****-20-**

Merritt teaches that a terminal authentication message be communicated to the user device along a separate connection between the user device and the server, without also communicating the message along the connection between the terminal and the server. Since that limitation is not taught or suggested by the cited references, and since that limitation is recited in all of the remaining pending claims, it cannot be concluded that the claims are rendered obvious by the combination of teachings of Merritt and Manduley.

The Examiner has stated that Manduley teaches that the "smartcard contains an LCD display that will, at the request of the server/issuing authority, display a message to the user", citing Col. 3, lines 11-16 and lines 47-58. However, displaying at the device/card is not sufficient to render the claims unpatentable. Even if one were to modify Merritt so that the user device could display the PSP, rather than the terminal displaying the PSP, one would not arrive at the invention as claimed.

The Examiner has cited the Hoss patent for its teachings related to sending a message along a first trusted connection between a terminal and a server. Applicants respectfully maintain that none of the cited patents teaches

SZ998-041

-21-

or suggests two different connections. None of the patents teaches or suggests providing a terminal authenticity message, and none teaches providing that message via an established second connection between the user device and the terminal without also communicating that message to the terminal along the first connection.

The Examiner has newly cited the Limsico patent for teaching that user passwords should be changed on a regular basis to increase the level of security of a user's account. The Examiner cites those teachings against the claimed "user-specific terminal authenticity message". Applicants respectfully assert that a terminal authenticity message has nothing to do with a user password. While the present invention can generate user-specific terminal authenticity messages, it is still a terminal authenticity message, and not a user password.

Applicants reiterate that none of the references teaches the claim features of establishing first and second trusted connections, of generating a terminal authenticity message in response to user input to a terminal, and of delivering the terminal authenticity message to the user device along the second trusted connection between the server and the user device without delivering the message to

SZ998-041

-22-

the terminal along the first trusted connection between the server and the terminal, and that a *prima facie* case of obviousness simply has not been presented by the Examiner. For a determination of obviousness, the prior art must teach or suggest all of the claim limitations. "All words in a claim must be considered in judging the patentability of that claim against the prior art" (*In re Wilson*, 424 F. 2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970)). If the cited references fail to teach each and every one of the claim limitations, a *prima facie* case of obviousness has not been established by the Examiner.

The addition of the teachings of the Schneier reference to the combination of Merritt and Manduley do not render the invention obvious. While Schneier can output a number to represent a message, there is nothing in Schneier which would lead one having skill in the art to modify the combination of Merritt and Manduley to include communication of terminal authentication along a connection between a server and a user device without delivery to a terminal along a connection between the terminal and the server.

The addition of the Lessin patent teachings to the combination of Merritt and Manduley does not render the pending claims obvious. Lessin has been cited for teaching

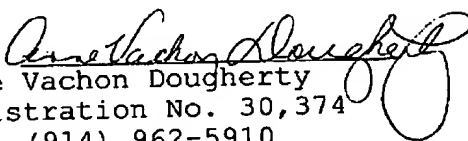
SZ998-041

-23-

user entry of a PIN. The combination of Merritt, Manduley, Hoss, Limsico and Lessin effectively teaches away from the claimed invention since the user would be forced to enter his PIN at a terminal before establishing that the terminal was trusted. Clearly that does not obviate the language of Claim 20, which expressly states that the server first send terminal authentication information directly to the user device and not the terminal for authenticating the user.

Based on the foregoing amendments and remarks, Applicants respectfully request entry of the amendments, reconsideration of the amended claim language in light of the remarks, withdrawal of the rejections, and allowance of the claims.

Respectfully submitted,  
N. Asokan, et al

By:   
Anne Vachon Dougherty  
Registration No. 30,374  
Tel. (914) 962-5910

SZ998-041

-24-



tunnelling from FOLDOC



tunnelling  
Random

[Search](#)[Home](#)[Contents](#)[Feedback](#)

## tunnelling

*<networking>* (US: "tunneling") Encapsulation of protocol A within protocol B, such that A treats B as though it were a data link layer. Tunnelling is used to get data between administrative domains which use a protocol that is not supported by the internet connecting those domains.

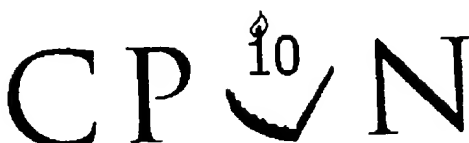
(1997-03-26)

Try this search on Wikipedia, OneLook, Google

**Nearby terms:** Tuki « tunafish « tune « **tunnelling** » TUPLE » tuple » tuple calculus

<http://foldoc.org/foldoc.cgi?tunnelling>

5/8/2006



[Home](#) · [Authors](#) · [Recent](#) · [News](#) · [Mirrors](#) · [FAQ](#) ·  
[Feedback](#)

in All

[CPAN Search](#)

[Philippe "BookK" Bruhat](#) > [connect-tunnel](#) > [connect-tunnel](#)

Source

[Download: connect-tunnel-0.03.tar.gz](#)

[Annotate this POD](#)

[NAME](#)  
[SYNOPSIS](#)  
[DESCRIPTION](#)  
[OPTIONS](#)  
[EXAMPLES](#)  
[ENVIRONMENT VARIABLES](#)  
[TODO](#)  
[AUTHOR](#)  
[COPYRIGHT](#)

## NAME

connect-tunnel - Create CONNECT tunnels through HTTP proxies

## SYNOPSIS

```
connect-tunnel [ -v ] [ -A user:pass ] [ -P proxy:port ] -T port:host:hostport [ -  
T port:host:hostport ]
```

## DESCRIPTION

**connect-tunnel** sets up tunneled connections to external hosts by redirecting connections to local ports towards those hosts/ports through a HTTP proxy.

**connect-tunnel** makes use of the HTTP **CONNECT** method to ask the proxy to create a tunnel to an outside server. Be aware that some proxies are set up to deny some outside tunnels (either to ports other than 443 or outside a specified set of outside hosts).

## OPTIONS

The program follows the usual GNU command line syntax, with long options starting with two dashes.

**-A, --proxy-authentication user:password**

Proxy authentication information.

<http://search.cpan.org/dist/connect-tunnel/connect-tunnel>

5/8/2006

connect-tunnel - Create CONNECT tunnels through HTTP proxies - search.cpan.org

Please note that all the authentication schemes supported by LWP::UserAgent are supported (we use an LWP::UserAgent).

This means we also support NTLM, since it is supported as from libwww-perl 5.66.

#### **-L, --local-only**

Create the tunnels so that they will only listen on `localhost`. Thus, only connections originating from the machine that runs **connect-tunnel** will be accepted.

That was the default behaviour in **connect-tunnel** version 0.02.

#### **-P, --proxy proxy[:port]**

The proxy is required to connect the tunnels. If no port is given, 8080 is used by default.

See also "ENVIRONMENT VARIABLES".

#### **-T, --tunnel port:host:hostport**

Specifies that the given *port* on the local host is to be forwarded to the given *host* and *hostport* on the remote side. This works by allocating a socket to listen to *port* on the local side, and whenever a connection is made to this *port*, the connection is forwarded through the proxy, and a connection is made to the remote *host* at port *hostport*.

On Unix systems, only root can forward privileged ports.

Note that you can setup tunnels to multiple destinations, by using the **--tunnel** option several times.

#### **-U, --user-agent string**

Specify User-Agent value to send in HTTP requests. The default is to send `connect-tunnel/version`.

#### **-v, --verbose**

Verbose output.

This option can be used several times for more verbose output.

## EXAMPLES

To connect to a SSH server running beyond the proxy on port 443, through the proxy `proxy.company.com`, running on port 8080, use the following command:

<http://search.cpan.org/dist/connect-tunnel/connect-tunnel>

5/8/2006

connect-tunnel - Create CONNECT tunnels through HTTP proxies - search.cpan.org

Page 3 of 4

```
connect-tunnel -P proxy.company.com:8080 -T 22:ssh.example.com:443
```

And now point your favorite ssh client to the machine running **connect-tunnel**.

You can also emulate a "standard" user-agent:

```
connect-tunnel -U "Mozilla/4.03 [en] (X11; I; Linux 2.1.89 i586)"  
-P proxy.company.com:8080 -T 22:ssh.example.com:443
```

**connect-tunnel** can easily use your proxy credentials to connect outside:

```
connect-tunnel -U "Mozilla/4.03 [en] (X11; I; Linux 2.1.89 i586)"  
-P proxy.company.com:8080 -T 22:ssh.example.com:443  
-A book:s3kr3t
```



But if you don't want anybody else to connect to your tunnels and through the proxy with *your* credentials, use the **-local-only** option:

```
connect-tunnel -U "Mozilla/4.03 [en] (X11; I; Linux 2.1.89 i586)"  
-P proxy.company.com:8080 -T 22:ssh.example.com:443  
-A book:s3kr3t -L
```

If you have several destinations, there is no need to run several instances of **connect-tunnel**:

```
connect-tunnel -U "Mozilla/4.03 [en] (X11; I; Linux 2.1.89 i586)"  
-P proxy.company.com:8080 -A book:s3kr3t -L  
-T 22:ssh.example.com:443  
-T 222:ssh2.example.com:443
```

But naturally, you will need to correctly set up the ports in your clients.

Mmm, such a long command line would perfectly fit in an alias or a **.BAT** file. ;-) )

## ENVIRONMENT VARIABLES

The LWP::UserAgent that is used to connect to the proxy accept the usual HTTP\_PROXY environment variable to define the proxy.

The environment variable is overridden by the **-proxy** option, if passed to **connect-tunnel**.

## TODO

<http://search.cpan.org/dist/connect-tunnel/connect-tunnel>

5/8/2006

Next version should have an option to create a control port, to which one could connect to interact with **connect-tunnel** and add/remove tunnels, close connections, change the User-Agent string, and so on.

## AUTHOR

Philippe "Book" Bruhat <book@cpan.org>

I seem to have re-invented a well-known wheel with that script, but at least hope I have added a few interesting options to it.

Bits of the documentation wording is stolen from OpenSSH documentation about options **-L** and **-R**.

## COPYRIGHT

This module is free software; you can redistribute it or modify it under the same terms as Perl itself.

---

hosted by perl.org, hardware provided by

